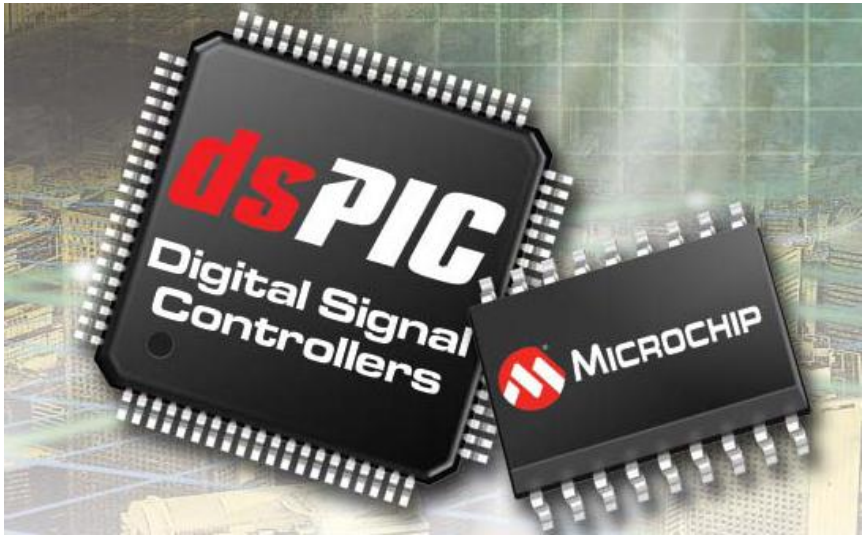
# EE427 Advanced Microcontrollers



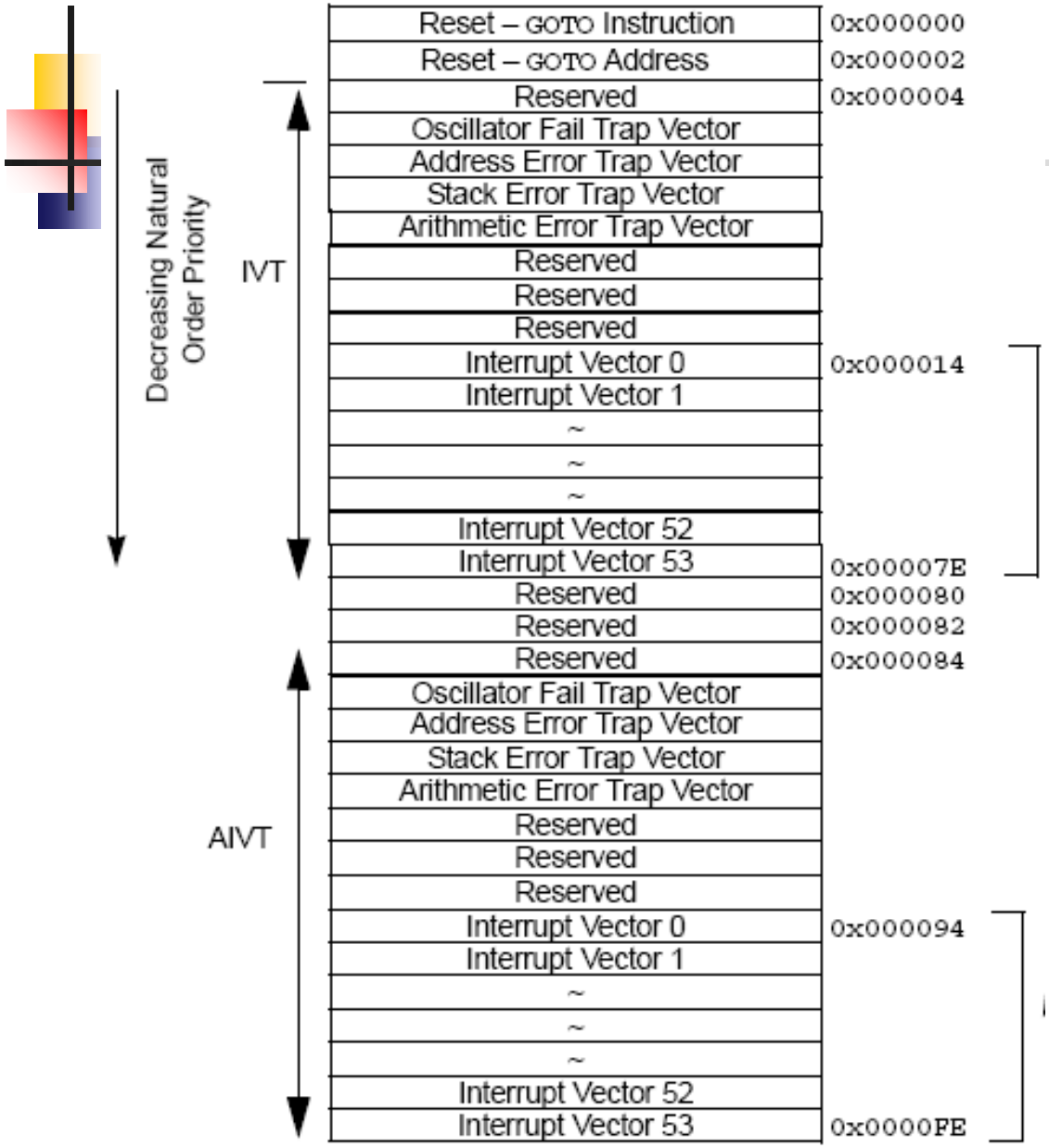
# Interrupts

# Interrupts

- Features include
  - Up to 8 processor exceptions and software traps
  - 7 user selectable priority levels
  - Interrupt Vector Table (IVT) with up to 62 vectors
  - A unique vector for each interrupt source
  - Fixed priority within a specified user priority level
  - Alternate Interrupt Vector Table (AIVT) for debug support

# Interrupt Vector Table (IVT)

- Resides in program memory, starts at location `0x000004`
- contains 62 vectors = 8 non-maskable trap vectors + 54 sources of interrupt
- In general, each interrupt source has its own vector
- Each interrupt vector contains a 24-bit wide address
- Value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR)


Decreasing Natural Order Priority
IVT
AIVT
Reset – GOTO Instruction
Reset – GOTO Address
Reserved
Oscillator Fail Trap Vector
Address Error Trap Vector
Stack Error Trap Vector
Arithmetic Error Trap Vector
Reserved
Reserved
Reserved
Interrupt Vector 0
Interrupt Vector 1
~
~
~
Interrupt Vector 52
Interrupt Vector 53
Reserved
Reserved
Reserved
Oscillator Fail Trap Vector
Address Error Trap Vector
Stack Error Trap Vector
Arithmetic Error Trap Vector
Reserved
Reserved
Reserved
Interrupt Vector 0
Interrupt Vector 1
~
~
~
Interrupt Vector 52
Interrupt Vector 53
0x000000
0x000002
0x000004
0x000014
0x00007E
0x000080
0x000082
0x000084
0x000094
0x0000FE

# Alternate Vector Table (AIVT)

- located after the IVT
- Access to the AIVT is provided by the ALTIVT control bit
- If ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors
- supports emulation and debugging efforts, provides a means to switch between application and support environment without requiring interrupt vectors to be reprogrammed

# Reset Sequence

- dsPIC30F device clears its registers in response to a Reset which forces the PC to zero
- Processor then begins program execution at location `0x000000`
- user programs **`GOTO`** instruction at Reset address which redirects program execution to appropriate start-up routine

# Trap Vector Details

| Vector Number | IVT Address | AIVT Address | Trap Source |
|---|---|---|---|
| 0 | 0x000004 | 0x000084 | Reserved |
| 1 | 0x000006 | 0x000086 | Oscillator Failure |
| 2 | 0x000008 | 0x000088 | Address Error |
| 3 | 0x00000A | 0x00008A | Stack Error |
| 4 | 0x00000C | 0x00008C | Arithmetic Error |
| 5 | 0x00000E | 0x00008E | Reserved |
| 6 | 0x000010 | 0x000090 | Reserved |
| 7 | 0x000012 | 0x000092 | Reserved |

# CPU Priority Status

- CPU can operate at one of sixteen priority levels, 0-15
- An interrupt or trap source must have priority level greater than the current CPU priority in order to initiate an exception process
- Peripheral and external interrupt sources can be programmed for level 0-7, while CPU priority levels 8-15 are reserved for trap sources
- Trap events have higher priority than any user interrupt source

# Non-Maskable Traps

- Traps can be considered as non-maskable, nestable interrupts which adhere to fixed priority structure and intended to provide the user a means to correct erroneous operation during debug and when operating within the application
- dsPIC30F has four implemented sources of non-maskable traps:
  - Oscillator Failure Trap
  - Stack Error Trap
  - Address Error Trap
  - Arithmetic Error Trap
- Oscillator failure trap has the highest priority, while arithmetic error trap has the lowest priority

# Soft and Hard Traps

- trap sources are classified as: ‘Hard’ traps and ‘Soft’ traps
- Soft traps can be treated like non-maskable sources of interrupt that adhere to the priority assigned by their position in the IVT.
- The arithmetic error trap and stack error trap are categorized as ‘soft’ trap sources

# Soft and Hard Traps

- Hard traps are also non-maskable sources of interrupt
- The address error and oscillator error traps are hard error traps
- hard traps force the CPU to stop code execution after the instruction causing the trap has completed
- Normal program execution flow will not resume until after the trap has been Acknowledged and processed

# Stack Error Trap

- stack is initialized to **`0x0800`** during Reset
- stack error trap will be generated when stack pointer address is less than `0x0800 and` over the end of data space (`0xFFFF`)

# Arithmetic Error Trap

- following events will cause an arithmetic error trap to be generated:
  - Accumulator A Overflow
  - Accumulator B Overflow
  - Catastrophic Accumulator Overflow
  - Divide by Zero
  - Shift Accumulator (SFTAC) operation exceeding +/-16 bits

# Overflow

- There are three enable bits in the INTCON1 register that enable the three types of accumulator overflow traps.
  - OVATE control bit (INTCON1<10>) is used to enable traps for an Accumulator A overflow event
  - OVBTE control bit (INTCON1<9>) is used to enable traps for an Accumulator B overflow event
  - COVTE control bit (INTCON1<8>) is used to enable traps for a catastrophic overflow of either accumulator
- An Acc A or Acc B overflow event is defined as a carry-out from bit 31.
- A catastrophic accumulator overflow is defined as a carry-out from bit 39 of either accumulator.

# Divide-by-zero and Shift ACC

## Divide-by-zero

- Divide-by-zero traps cannot be disabled.
- Divide-by-zero check is performed during the first iteration of the **`REPEAT`** loop that executes the divide instruction.

## Shift Accumulator

- Accumulator shift traps cannot be disabled.
- **`SFTAC`** instruction can be used to shift the accumulator by a literal value or a value in one of the W registers.
- If the shift value exceeds +/-16 bits, an arithmetic trap will be generated. The `SFTAC` instruction will execute, but the results of the shift will not be written to the target accumulator.

# Trap Priority and Hard Trap conflicts

- If a higher priority trap occurs while any lower priority trap is in progress, processing of the lower priority trap will be suspended and the higher priority trap will be Acknowledged and processed.
- The lower priority trap will remain pending until processing of the higher priority trap completes.

- If a lower priority hard trap occurs while a higher priority trap is pending, Acknowledged, or is being processed, a hard trap conflict will occur.
- The conflict occurs because the lower priority trap cannot be Acknowledged until processing for the higher priority trap completes.

- The device is automatically reset in a hard trap conflict condition.

# Oscillator Failure Trap

- An oscillator failure trap event will be generated for any of the following reasons:
  - Fail-Safe Clock Monitor (FSCM) is enabled and has detected a loss of the system clock source.
  - A loss of PLL lock has been detected during normal operation using the PLL.
  - FSCM is enabled and the PLL fails to achieve lock at a Power-On Reset (POR).

# Address Error Trap

- causes for an address error trap to be generated:
  - A misaligned data word fetch is attempted. This condition occurs when an instruction performs a word access with the LSb of the effective address set to ‘1’.
  - A bit manipulation instruction using the Indirect Addressing mode with the LSb of the effective address set to ‘1’.
  - A data fetch from unimplemented data address space is attempted.
  - Execution of a “**`BRA #literal`**” instruction or a “**GOTO #literal**” instruction, where literal is an unimplemented program memory address.
  - Executing instructions after modifying the PC to point to unimplemented program memory addresses.

- Topic for next session
  - Disable interrupt instruction
  - Interrupt processing timing
  - Interrupt control registers